VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Metody výběru atributů

Feature Selection Methods

2014 Tomáš Bystřický

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Tomáš Bystřický**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Metody výběru atributů
Feature Selection Methods

Zásady pro vypracování:

Cílem práce je udělat přehled algoritmů pro výběr atributů (feature selection) a implementovat jednu nebo více metod. Implementované metody budou otestovány nad praktickými příklady. Práce bude obsahovat:

1. Přehled metod pro výběr atributů
2. Podrobný popis zvolené/zvolených metod.
3. Návrh implementace zvolených metod.
4. Experimentální ověření zvolených metod nad reálnými a/nebo vygenerovanými daty.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jan Platoš, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 07.05.2014

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne 18. 7. 2014 

Poděkování

Chtěl bych poděkovat vedoucímu mé diplomové práce doc. Ing. Janu Platošovi, Ph.D. za cenné rady a připomínky k danému tématu a za pomoc při koncipování textové části práce. Rovněž mu děkuji za vstřícnost a ochotu.

Také bych chtěl poděkovat své rodině a nejbližšímu okolí a to především za psychickou podporu.

Abstrakt

Cílem této diplomové práce je vytvořit přehled algoritmů, které se používají pro selekci atributů. Dále pak vybrané algoritmy naimplementovat.

Pro ověření funkčnosti algoritmů a zjištění jejich úspěšnosti v návaznosti na klasifikaci dat si buď vygenerovat vlastní datové soubory, nebo získat reálné datové soubory z internetu či jiného zdroje.

Nakonec práce pak zhodnotit výsledky z provedených experimentů.

Klíčová slova: výběr atributů, chí-kvadrát test, Pearsonův korelační koeficient, Kruskal-Wallis test, klasifikace, Weka

Abstract

The aim of this diploma thesis is to create overview feature selection methods. Then, the selected algorithms implement.

For verify of function algorithms and finding their successfulness to classification data either generate own datasets or obtain real datasets from internet or other resource.

In the end work to evaluate the results of the experiments.

Keywords: feature selection, chi-square test, Pearson correlation coefficient, Kruskal-Wallis test, classification, Weka

Seznam použitých symbolů a zkratek

FS	Feature Selection, výběr atributů
csv	datový formát souboru s hodnotami oddělenými čárkami, případně jiným znakem
xlsx	datový formát aplikace Microsoft Excel

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Každým dnem vzniká ve firmách, výzkumných organizacích a dalších institucích obrovské množství dat. Se stále se zvyšujícími kapacitami datových úložišť ale nikdo neřeší, která data uchovávat a která ne. Vzniká tak stále více dat, ovšem informační hodnota v nich nestoupá, tak jak by se mohlo zdát. V takovýchto kvantech informací jsou ukládána data nadbytečná, tedy vzájemně závislá nebo redundantní. Bohužel tento fakt si zprvu neuvědomujeme. Až pak, když s těmito daty začneme pracovat, hledat v nich informace a souvislosti, třídit je, zjistíme, že použité postupy a algoritmy selhávají. I na ně je spousta informací jednoduše moc. Pak nedostáváme tak přesné výsledky, jak jsme očekávali.

Cílů této práce je několik. Zaprvé vytvořit přehled metod, které se snaží v těchto datech nalézt ta, která pokud dojde k jejich odstranění, nijak se nesníží informační hodnota celého datového souboru. Druhým cílem bude vybrané metody naimplementovat a vyzkoušet, zda se chovají tak, tak jsme očekávali. Bude však nutné připravit datové soubory. Některá data si vygeneruji sám a jiná si zase obstarám z internetu. Reálná data mají tu schopnost, že se obvykle chovají trochu jinak, než data vygenerována. Dalo by říci, že nás vždy něčím překvapí. Provedu tedy experimenty jak na vygenerovaných, tak na reálných datech. A konečně, jako poslední krok, zhodnotím výsledky mých experimentů.

2 Redukce dimenzionality

Snížení dimenzionality je podstatné v mnoha oblastech zpracování dat. Jsou to např. databáze, systémy strojového učení apod. Nabízí nám neocenitelné výsledky např. u vizualizace těchto dat nebo také zvyšuje přesnost klasifikace. Přináší rychlé a efektivní načítání dat, jejich kompresi apod.

Jednotlivá data jsou uložena v databázích. Jsou popsána pomocí atributů např. jméno, věk, bydliště, atd. Předpokládejme tedy, že původní datový soubor obsahuje *N* takovýchto atributů. Úkolem redukce dimenzionality je vytvořit *K* atributů, *kde* $K < N$. V principu jsou dvě možnosti jak snížit počet atributů v datech. Je to „selekce atributů" a „extrakce atributů" [1]

2.1 Selekce atributů

Cílem „selekce atributů" nebo také „feature selection" metod je zanechání podmnožiny *K* nejlepších atributů z celkového počtu *N* atributů a ostatní atributy odstranit. Tyto metody nijak nemění původní atributy. Metody lze rozdělit na „filters", „wrappers" a „embedded". Právě těmto metodám se budu v práci věnovat. [1]

2.2 Extrakce atributů

Extrakce atributů je také známá jako „transformace atributů". Je to proces, který najde nový počet dimenzí *K*, které jsou kombinací *N* původních dimenzí. Nejznámější tyto techniky jsou založeny na projekčních a kompresních metodách. Jako příklad můžeme jmenovat metody „Principal component analysis" (PCA) – „Analýza hlavních komponent" a „Linear discriminant analysis" (LDA) – „Lineární diskriminační analýza". [1]

3 Klasifikace dat

Klasifikace je proces „data miningu“ (dolování z dat), kdy dochází k třídění informací do kategorií. Neformálně se dá také říci, že data z jedné třídy jsou si podobnější, než data z různých tříd. [2]

Tématem této práce není klasifikace dat, proto nebudu toto téma rozvádět do větších detailů. Avšak je to jedna z technických disciplín, kde se uplatňuje využití snižování dimenzionality dat.

V praktické části této práce používám ke klasifikaci dat aplikaci Weka. V ní využívám několik klasifikátorů, abych mohl porovnat úspěšnost klasifikace před a po odstranění atributů z datového souboru. Nyní krátce představím aplikaci Weka a vytvořím přehled použitých klasifikátorů.

3.1 Weka

Weka je aplikace využívána v oblastech strojového učení. Je naprogramována v Javě a její název je zkratkou názvu „Waikato Environment for Knowledge Analysis“. Pochází z Nového Zélandu z University of Waikato. Historie této aplikace sahá až do roku 1993 [3]. Na jejím logu je umístěn pták weka, vyskytující se rovněž na tomto území.

Po funkční stránce nabízí Weka např. klasifikaci dat, klastrování, hledání asociací a další funkce. Pro klasifikaci dat nabízí přes 100 klasifikačních metod.

3.1.1 J48

J48 je klasifikátor ze skupiny „trees“. Jedná se o implementaci ID3 (Iterative Dichotomiser 3) algoritmu, který vytvořil Ross Quinlan. Při své práci generuje rozhodovací strom. [4] [5]

Lze použít pro klasifikaci dat, která jsou pouze diskrétní.

3.1.2 SimpleCart

Další ze skupiny „trees“ metod, který má stejné požadavky jako klasifikátor J48. Vyžaduje tedy pouze diskrétní data.

3.1.3 BayesNet

Klasifikátor BayesNet již patří do skupiny tzv. „bayesovských“ klasifikátorů. Tzn., že je založen na použití Bayesovy věty. Bayesova věta je věta z teorie pravděpodobnosti – pochází od Thomase Bayese. Věta nám říká, jak podmíněná pravděpodobnost nějakého jevu souvisí s opačnou podmíněnou pravděpodobností. [6]

Používá se pro klasifikaci diskrétních dat.

3.1.4 SMOreg

Metoda implementuje sekvenční minimální optimalizační algoritmus pro trénování a podporu vektorového regresního modelu. Tento algoritmus pochází od Alexe Smoly a Bernharda Scholkopfa. [7]

Jedná se o klasifikátor ze skupiny „function“. Je schopen pracovat se spojitými i diskrétními daty. Klasifikuje pouze podle atributu se spojitými daty.

3.1.5 DecisionTable

DecisionTable metoda patří do skupiny „rules“ metod. Je schopna pracovat jak se spojitými, tak s diskrétními daty.

3.1.6 M5Rules

Poslední klasifikační metoda, kterou používám v této práci, patří rovněž do kategorie „rules“. Metoda generuje rozhodovací seznam pro regresní problémy za použití metody "rozděl a panuj". V každé iteraci tvoří algoritmus strom a přidává "nejlepší" list do seznamu pravidel. [8]

Pracuje s daty diskrétními i spojitými. Klasifikační atribut však musí být spojitého charakteru.

Při použití v praktické části práce uvádím jednotlivé výsledky klasifikace. U diskrétních (kategoriálních) dat poskytují metody procentuální vyjádření správně klasifikovaných instancí. Pokud jsou data spojitá, výstupem metody je tzv. relativní absolutní chyba.

4 Přehled algoritmů pro selekci atributů

4.1 Metody typu Filters

Jak napovídá název, metody typu Filters jsou algoritmy, které odfiltrovávají nepodstatné atributy. To jsou ty, které mají jen malou šanci být užitečnými při další analýze dat. Filters metody jsou výpočetně méně náročné než metody uváděné v dalších kapitolách. Jejich nevýhodou je, že mají tendenci odstranit malé množství závislých atributů. Proto je potřeba správně nastavit práh pro vybrání podmnožiny atributů. [1]

4.1.1 Chi-square test

Také Chi-squared test, Chi-kvadrát test nebo Test dobré shody. Tento test se používá pro ověření, zda náhodný výběr pochází z určitého rozdělení, např. normálního. Je také schopen odhalit závislost mezi pozorovanými a odhadovanými četnostmi ve výběrovém souboru. V tomto případě očekává kategoriální data.

Pokud máme data spojitá, je nutné rozdělit je do intervalů. Pokud to lze, je lepší rozdělit data na konstantní intervaly. Není to však podmínkou. Počet intervalů volíme tak, aby nebyl ani příliš malý ani příliš velký. Malý počet intervalů vede ke zjednodušenému pohledu na rozdělení pravděpodobnosti. Příliš velký počet intervalů nám zase dělá rozdělení pravděpodobnosti nepřehledným. Obvykle je tedy vhodné volit 5 – 15 intervalů.

Nutnou podmínkou provedení tohoto testu je, aby všechny očekávané četnosti byly větší než 5. Pokud tento předpoklad není splněn, lze intervaly sloučit.

Testem ověřujeme tzv. nulovou hypotézu. Oproti nulové hypotéze definujeme hypotézu alternativní. Ta je vždy negací nulové hypotézy a platí tehdy, když neplatí hypotéza nulová.

Jako testovou statistiku volíme statistiku G, která má asymptoticky χ^2_{k-h-1} rozdělení:

$$G = \sum_{i=1}^{k} \frac{\left(n_i - n \cdot \pi_{0,i}\right)^2}{n \cdot \pi_{0,i}} \rightarrow \chi^2_{k-h-1}$$

kde n je rozsah výběru, k je počet variant, h je počet odhadovaných parametrů modelu, n_i jsou skutečné četnosti variant a $\pi_{0,i}$ jsou očekávané relativní četnosti. V případě, že je splněná nulová hypotéza, tak by právě těchto hodnot měly nabýt jednotlivé skutečné četnosti.

U testu může dojít k chybě 1. a 2. druhu.

- Chyba 1. druhu – nulová hypotéza sice platí, ale my ji zamítáme. U testu volíme tzv. hladinu významnosti α, což je max. možná pravděpodobnost chyby 1. druhu.
- Chyba 2. druhu – vyskytne se, pokud nulová hypotéza neplatí, ale my ji nezamítáme. Značí se β.

[9]

Pro praktické využití chi-square testu pro selekci atributů, uvedu tento příklad. Algoritmus pro určení závislých atributů lze rozdělit do 6 kroků. Budeme předpokládat datový soubor, jehož vzorek ukazuje Obrázek 1.

V1 = Income Class	V2 = # of children	V3 = Number of bedrooms	V4 = job type
L	2 or 3	2	Hourly
L	more than 3	3	Contract
U	2 or 3	5	Business
M	2 or 3	4	Professional
M	1	4	Professional
L	more than 3	2	Contract
U	2 or 3	4	Professional
M	2 or 3	3	Professional
M	2 or 3	4	Contract
M	0	3	Contract
M	2 or 3	4	Professional

Obrázek 1: Ukázka datového souboru [10]

Krok 1 – Zjištění tříd u každého atributu

V prvním kroku algoritmus projde všechny záznamy tabulky a u každého atributu identifikuje unikátní položky. Tyto unikátní položky budeme nazývat třídy. V našem případě V1 = 3 třídy, V2, V3, V4 = 4 třídy.

Krok 2 – Sestavení kontingenčních tabulek

Ve druhém kroku algoritmus sestaví pro každé dva atributy kontingenční tabulku. Kontingenční tabulka je tabulka, kde označení sloupců tvoří třídy prvního atributu a označení řádků pak třídy druhého atributu. Každá hodnota v této tabulce pak představuje počet výskytů

záznamů v původní tabulce pro odpovídající označení řádek × sloupec. Takže jak ukazuje Obrázek 2, počet případů v původní tabulce, kdy V1 = L a zároveň V2 = 0 bylo 7.

		Income Class = C1 columns		
		L	M	U
# of children = C2 rows	0	7	18	6
	1	9	38	23
	2 or 3	34	97	58
	more than 3	47	31	30

Obrázek 2: Kontingenční tabulka pro atributy V1 a V2 [10]

Počet kontingenčních tabulek, které bude potřeba sestavit, se dá vyjádřit následujícím vztahem.

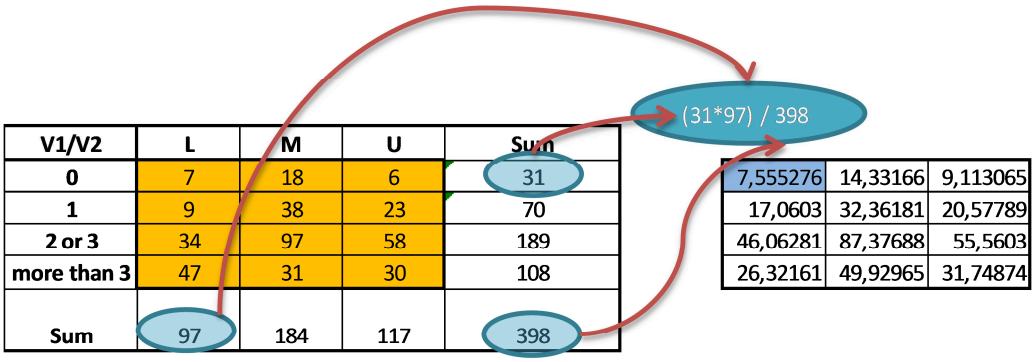
$$počet\ kontingenčních\ tabulek = \frac{n * (n-1)}{2}$$

Kde n je počet sloupců původní tabulky. Je tedy vidět, že počet kontingenčních tabulek roste kvadraticky. V našem případě pro 4 sloupce bude potřeba sestavit 6 kontingenčních tabulek. Dále se v popisu algoritmu omezím pouze na první kontingenční tabulku.

Krok 3 – Sestavení tabulek očekávaných četností

Když už máme kontingenční tabulky, je potřeba sestavit tzv. tabulky očekávaných četností. Těchto tabulek bude stejně jako kontingenčních tabulek. Rovněž budou mít stejný počet řádků a sloupců.

Tyto tabulky se vytvoří tak, že u všech řádků a sloupců vypočteme jejich součty. Taktéž spočteme celkový součet tabulky. Jednotlivé hodnoty tabulky spočteme jako podíl součinu dvou odpovídajících součtů a součtu celkového. Názorně to ukazuje Obrázek 3.

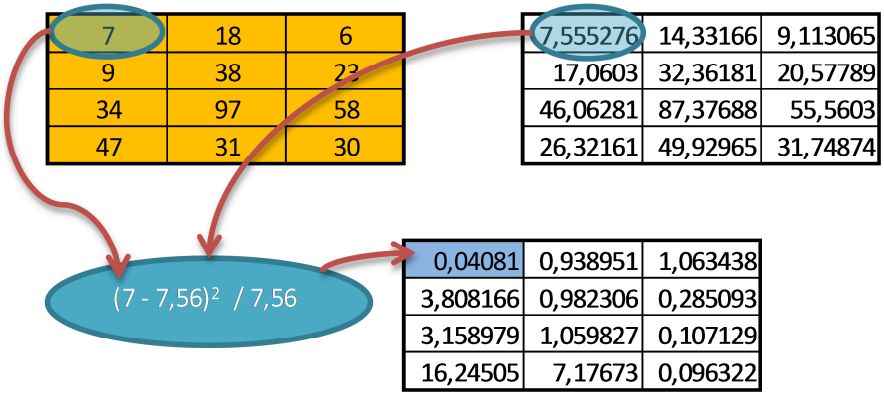


Obrázek 3: Sestavení tabulky očekávaných četností

Pokud nastane případ, že by některé očekávané četnosti byly menší než 5 (nutný předpoklad testu), dotčené řádky sloučíme. Odpovídající řádky sloučíme i v kontingenční tabulce. Jestliže by došlo ke sloučení všech řádků do jednoho, tak v tomto případě nelze test provést.

Krok 4 – Výpočet chi-square hodnoty

Ve 4. kroku již spočteme chi-square hodnotu. K výpočtu je potřeba kontingenční tabulku a tabulku očekávaných četností. Dílčí hodnoty vypočteme jako podíl druhé mocniny rozdílu odpovídajících hodnot a odpovídající hodnoty očekávané četnosti. Názorně to ukazuje Obrázek 4. Chi-square hodnota je pak dána součtem všech těchto dílčích hodnot.



Obrázek 4: Výpočet chi-square hodnoty

Pro tuto kontingenční tabulku je chi-square hodnota 34,96.

Krok 5 – Určení stupně volnosti

Pro pozdější zamítnutí nebo nezamítnutí nulové hypotézy je potřeba znát tzv. stupeň volnosti. Ten je dán následujícím vzorcem.

$$stupeň\ volnosti = (počet\ řádků\ -\ 1) * (počet\ sloupců\ -\ 1)$$

V našem případě vychází stupeň volnosti 6.

Krok 6 – Určení závislosti

Posledním krokem je určení, zda jsou na začátku zmiňované atributy V1 a V2 závislé. Použijeme standardní tabulky chi-square rozdělení, např. z [11] a vybereme si hladinu významnosti. V tomto případě jsem zvolil běžně používanou hladinu významnosti 0,5. Nakonec si stanovíme nulovou a alternativní hypotézu.

H_0: Mezi atributy neexistuje závislost (vypočtená hodnota < kritická hodnota)

H_A: Atributy jsou závislé.

- Vypočtená hodnota – 34,96
- Stupeň volnosti – 6
- Kritická hodnota – 12,592

Protože je vypočtená hodnota větší než hodnota kritická, zamítáme nulovou hypotézu ve prospěch hypotézy alternativní. Tedy mezi atributy V1 a V2 existuje závislost. Proto můžeme jeden z těchto dvou atributů odstranit.

Předchozí kroky opakujeme pro všechny kombinace atributů, a pokud je zjištěna závislost, vždy jeden z atributů odstraníme. [10] [12]

4.1.2 Pearsonův korelační koeficient

Výpočet Pearsonova korelačního koeficientu *r* se používá pro zjištění vztahu mezi dvěma spojitými veličinami. Ukazuje nám míru, jak moc jsou tyto veličiny závislé. Koeficient nabývá hodnot z intervalu -1 pro úplnou zápornou korelaci a +1 pro úplnou kladnou korelaci.

Hodnota *r* je kladná, když vyšší hodnoty *x* souvisí s vyššími hodnotami *y*. Když nižší hodnoty *x* souvisí s vyššími hodnotami *y*, je hodnota *r* záporná.

Pro hodnotu koeficientu rovnu -1 nebo 1 lze vztah mezi těmito veličinami vyjádřit pomocí funkčního lineárního vztahu. Při hodnotě 0 (nebo hodnotě okolo 0) tohoto koeficientu lineární vztah vylučujeme. Neznamená to však, že mezi veličinami neexistuje závislost.

Dá se taky říci, že tento koeficient charakterizuje variabilitu kolem lineárního trendu.

Koeficient se vypočte pomocí následujícího vztahu.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

kde $\bar{x}$ a $\bar{y}$ tvoří výběrové průměry

Pro hodnotu *r* platí následující

- Zanedbatelný vztah $r < 0{,}2$
- Nepříliš těsný vztah $r = 0{,}2 – 0{,}4$
- Středně těsný vztah $r = 0{,}4 – 0{,}7$
- Velmi těsný vztah $r = 0{,}7 – 0{,}9$
- Extrémně těsný vztah $r > 0{,}9$

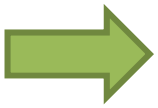
[13] [14] [15] [16]

Při selekci atributů pak algoritmus pracuje tak, že pro každé dva atributy načte jejich hodnoty a vypočte Pearsonův korelační koeficient. Protože nás zajímá pouze velikost, tak vypočte absolutní hodnotu. Tu pak porovná se zvoleným parametrem. Ten se zvolí na začátku experimentu. Pokud je vypočtená hodnota vyšší než zvolený parametr, tak jeden z atributů algoritmus odstraní. Výstupem je pak datový soubor bez těchto závislých atributů.

4.1.3 Kruskal-Wallis test

Kruskal-Wallis test je neparametrickou obdobou jednofaktorové ANOVy. Nepředpokládá normalitu dat, na rozdíl od ANOVy má však nižší citlivost. Tento test je vícevýběrovým testem mediánů.

Výběr	
1.	2.
11	8
6	-3
1	-1
13	2



Výběr	
1.	2.
7	6
5	1
3	2
8	4

Obrázek 5: Transformace tabulky

V prvním kroku algoritmus provede záměnu hodnot za jejich pořadí, jak ukazuje Obrázek 5.

Následně provede součty pořadí pro jednotlivé výběry. My máme 2 výběry proto:

$T_1 = 23$

$T_2 = 13$

Testová statistika Q je pak definována následovně:

$$Q = \frac{12}{N \cdot (N+1)} \cdot \sum_{i=1}^{k} \frac{T_i^2}{n_i} - 3 \cdot (N+1) \rightarrow \chi^2_{k-1}$$

kde N – celkový rozsah (počet hodnot)

T_i – součet pořadí pro i-tý výběr

n_i – rozsah i-tého výběru

Testová statistika se řídí χ^2_{k-1} rozdělením, kde k je počet výběrů.

Nakonec algoritmus porovná vypočtenou hodnotu Q s příslušnou hodnotou v tabulce kritických hodnot chi-square rozdělení na zvolené hladině významnosti.

Jako uvedené výběry lze chápat hodnoty jednotlivých atributů. Takto lze algoritmus použít na nekategoriální data.

Druhou možností je použití na kombinaci atributů kategoriálních a nekategoriálních. S tím, že jednotlivé výběry budou tvořit hodnoty podle kategorií, viz Obrázek 6.

a	6
a	-1
a	8
a	16
b	-6
b	3
b	7
b	0
c	11
c	24
c	-4
c	7

Obrázek 6: Kategoriální × nekategoriální data

[9]

4.1.4 ANOVA

Tento test se zaměřuje na analýzu rozptylů mezi několika třídami. Jde o zkratku z „ANalysis Of VAriance“. Test předpokládá normalitu rozdělení a tzv. homoskedasticitu, tedy identické rozptyly. Pokud nejsou tyto podmínky testu splněny, nelze test použít. Lze však aplikovat předešlý Kruskal-Wallis test, který je obdobou jednofaktorového třídění u ANOVy. Jeho nevýhodou je ale menší citlivost.

Nejprve je nutné zjistit tzv. rozptyl mezi třídami. Ten je dán následujícím vztahem.

$$S_B^2 = \frac{1}{k-1} \cdot \sum_{i=1}^{k} \left(\overline{X_i} - \overline{X}\right)^2$$

kde k – je počet tříd

n – je počet pozorování v jednotlivých třídách

$N = \sum_{i=1}^{k} n$ – je celkový počet pozorování

$\overline{X_i}$ – je průměr i-tého náhodného výběru

$\overline{X}$ – je celkový průměr

Rozptyl mezi třídami však neposkytuje dostatečnou informaci, neboť nezaznamenává kolísání v jednotlivých třídách. Zavádíme proto ještě rozptyl uvnitř tříd. Definován je následovně.

$$S_W^2 = \sum_{i=1}^{k} \frac{\sum_{j=1}^{n} (X_{ij} - \overline{X_i})^2}{n-1}$$

Nakonec vypočteme tzv. F-poměr. Vypočte se podle následujícího vztahu.

$$F - ratio = \frac{n \cdot S_B^2}{S_W^2}$$

F-poměr má Fisher-Snedecorovo rozdělení s počtem stupňů volnosti:

-pro čitatele (*k-1*)

-pro jmenovatele (*N-k*)

Zvolíme si tedy hladinu významnosti a v tabulkách pro Fisher-Snedecorovo rozdělení dle stupňů volnosti odečteme hodnotu. Tu porovnáme s F-poměrem. Jestliže je F-poměr nižší než odečtená hodnota, závislost mezi třídami neexistuje. V opačném případě jsou třídy závislé.

[9]

4.1.5 Information gain

Information gain nebo také „informační zisk“ a poměrný informační zisk jsou míry odvozené z entropie. Jsou založeny na principu minimalizace entropie. Informační zisk měří redukci entropie způsobenou volbou atributu *A*. Vypočte se pomocí tohoto vzorce:

$$Zisk(A) = H(C) - H(A)$$

Entropie nám vyjadřuje míru neuspořádanosti nebo neurčitosti nějakého systému. Abychom zjistili informační zisk atributu, musíme zjistit *H(A)* a *H(C)*. Výpočet *H(A)* je dán následujícími vztahy:

$$H(A) = \sum_{j=1}^{n} P(a_j) H(a_j)$$

$$H(a_j) = -\sum_{i=1}^{m} P(c_i \mid a_j) \log\left(P(c_i \mid a_j)\right)$$

kde A – atribut

a_j – jsou hodnoty tohoto atributu

c_i – jsou klasifikační třídy

P – pravděpodobnost

Logaritmus ve výše uvedeném vztahu je o základu větším než 1. Obvykle se jako základ volí hodnota 2. Pokud je pravděpodobnost nulová, tak se taky výraz *P*log(P)* bude rovnat 0.

H(C) pak vypočteme jako:

$$H(C) = -\sum_{i=1}^{m} P(c_i)\log\left(P(c_i)\right)$$

Informační zisk a entropie nebere v úvahu počet hodnot daného atributu. Proto byl zaveden tzv. poměrný informační zisk. Je dán těmito vztahy.

$$PoměrnýZisk(A) = \frac{Zisk(A)}{Větvení(A)}$$

$$Větvení(A) = -\sum_{j=1}^{n} P(a_j)\log\left(P(a_j)\right)$$

[17], [18]

4.1.6 Mutual information

Známá také pod názvem „vzájemná informace“. Určuje míru vzájemné informace mezi náhodnými proměnnými. Je dána vztahem:

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

kde *p(x,y)* – je sdružená distribuční funkce proměnných *X* a *Y*

p(x) resp. *p(y)* jsou marginální distribuční funkce proměnných *X* resp. *Y*

Vzájemná informace je dobře známá nelineární míra statistické závislosti založena na informační teorii. Je vždy nezáporná a také symetrická. Tedy *I(X;Y) = I(Y;X)*. Tento vztah se rovná nule, právě tehdy když *X* a *Y* jsou statisticky nezávislé. Tím se myslím, že *X* neobsahuje žádnou informaci o *Y*. [1]

4.1.7 Odds Ratio

Tato metoda byla původně navržena pro zpětnou vazbu relevance v klasifikaci textu. Základní myšlenka je, že rozdělení atributů v relevantních dokumentech je rozdílné od rozdělení atributů v nerelevantních dokumentech. Je definována takto:

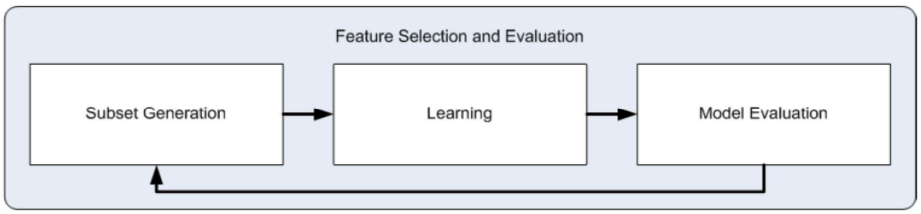
$$OR(f_k, c_i) = \frac{Pr(f_k|c_i) \times (1 - Pr(f_k|c_i))}{(1 - Pr(f_k|c_i)) \times Pr(f_k|c_i)}$$

[19]

4.2 Metody typu Wrappers

Metody spadající pod tento typ jsou chápány jako černá skřínka – nejsou potřeba žádné znalosti o algoritmu, důležité je pouze rozhraní. [20]

Tyto metody vyberou podmnožinu atributů použitím učících algoritmů. A to jako součást ohodnocovací funkce, jak ukazuje Obrázek 7.



Obrázek 7: Princip metod typu Wrappers [1]

Učící algoritmus se používá jako druh černé skřínky. Ohodnocovací funkce pro každého kandidáta na množinu atributů vrátí odhad kvality modelu, který je indukován učícím algoritmem. Ten proto způsobuje lepší odhad přesnosti.

Wrappers metody mají tendenci být pomalejší a výpočetně drahé, protože je během hledání neustále spouštěn učící algoritmus. [1]

4.2.1 Algoritmus sekvenční dopředné selekce (SFS)

Jedná se o jednoduchý algoritmus, který hledá postupně optimální řešení. Algoritmus začíná s prázdnou množinou, do které přidá atribut s nejlepší informační (selekční) hodnotou. Přidání tohoto atributu nám sníží chybu klasifikace ze všech možných atributů nejvíce. V každém dalším kroku pak přidává algoritmus atributy, které s již dříve přidanými atributy dosáhly nejlepší hodnoty kritéria dle vzorce:

$$J(\{X_{k+1}\}) = \max J\left(\{X_k \cup y_j\}\right)$$
$$y_j \epsilon \{Y - X_k\}$$

Tento algoritmus pracuje maximálně v n-rozměrném prostoru, proto je výpočetně jednodušší.

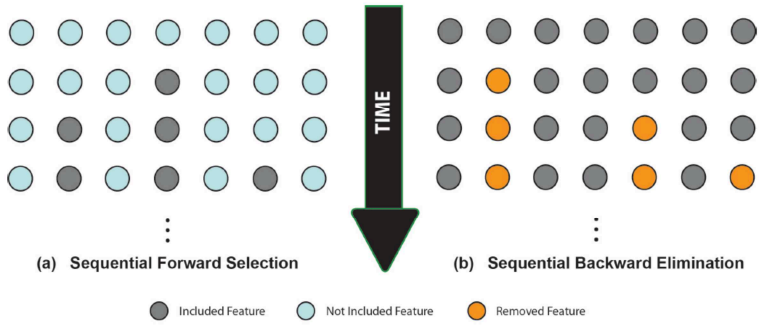
Suboptimalita nalezeného řešení je způsobena tím, že nelze vyloučit ty atributy, které se staly nadbytečné po přiřazení dalších veličin. [21]

4.2.2 Algoritmus sekvenční zpětné selekce (SBS)

Na rozdíl od předešlého algoritmu, tento algoritmus začíná s množinou, ve které jsou obsaženy všechny atributy. V každém dalším kroku odstraní algoritmus z množiny ten atribut, který způsobuje nejmenší pokles kriteriální funkce. Takže po kroku *k+1* platí:

$$J(\{X_{m-k-1}\}) = \max J\left(\{X_{m-k} - y_j\}\right)$$
$$y_j \epsilon \{X_{m-k}\}$$

Na rozdíl od předešlého algoritmu, tento algoritmus má tu výhodu, že může průběžně sledovat množství ztracené informace. Neexistuje však možnost opravy při neoptimálním vyloučení některého z atributů. [21]



Obrázek 8: Ukázka SFS a SBS algoritmů [1]

4.2.3 Zobecněný algoritmus sekvenční dopředné selekce (GSFS)

Algoritmus je zobecněním SFS algoritmu. Začíná rovněž s prázdnou množinou a opakovaně přidává podmnožinu s největší informační hodnotou. Ta je nalezena důkladným vyhledáváním. [22]

4.2.4 Zobecněný algoritmus sekvenční zpětné selekce (GSBS)

Algoritmus GSBS zobecňuje algoritmus SBS. Taktéž začíná s množinou, kde jsou obsaženy všechny atributy. Opakovaně odebírá nejméně významnou podmnožinu atributů. [22]

4.2.5 Algoritmus Plus p mínus q

Princip algoritmu je ten, že pokud přidá p atributů, tak odstraní q atributů. Algoritmus běží tak dlouho, dokud se nedosáhne určeného počtu atributů. Pokud:

$p > q$ algoritmus pracuje od prázdné množiny

$p < q$ algoritmus pracuje od množiny se všemi atributy

[21]

4.2.6 Algoritmus Min – Max

Je heuristický algoritmus, který vybírá atributy na základě výpočtu hodnot kriteriální funkce pouze v jedno- a dvourozměrném atributovém prostoru.

Uvažujme tedy, že bylo vybráno k příznakových veličin do množiny $\{X_k\}$ a zbývají veličiny z množiny $\{Y - X_k\}$. Výběr veličiny $y_j \in \{Y-X_k\}$ přináší novou informaci, kterou můžeme ocenit relativně k libovolné veličině $x_i \in X_k$ podle vztahu:

$$\Delta J(y_j,x_i) = J(y_j,x_i) - J(x_i)$$

Informační přírůstek ΔJ pak musí být co největší, ale zároveň musí být dostatečný pro všechny veličiny již zahrnuté do množiny X_k. Vybereme si tedy pak veličinu $y_k + 1$, pro kterou platí:

$$\Delta J(y_k+1,x_k) = max_j\, min_i\, \Delta J(y_j,x_i),\ x_i \in X_k$$

[21]

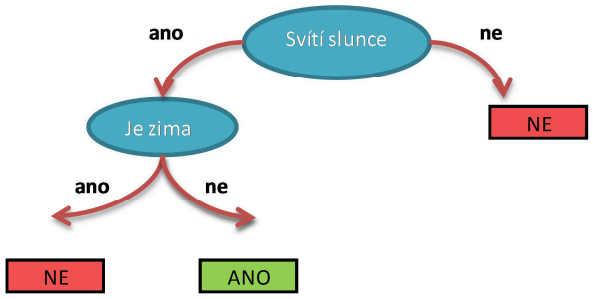
4.3 Embedded metody

Na rozdíl od metod typu Filters a Wrappers, učící část a část týkající se výběru atributů se provádí společně v metodě klasifikace. [1]

Klasifikace není tématem této práce, proto uvedu jen pro ukázku jeden příklad takovéhoto algoritmu.

4.3.1 Rozhodovací stromy

Reprezentování znalostí v podobě stromu se běžně používá v celé řadě oblastí. Používáme je i v běžném životě, ač o tom ani nevíme. Obrázek 9 např. ukazuje rozhodovací strom, zda jít ven na procházku nebo ne.



Obrázek 9: Rozhodovací strom

Při tvorbě rozhodovacího stromu se postupuje metodou „rozděl a panuj“. Trénovací data se postupně rozdělují na menší a menší podmnožiny, tedy uzly stromu, tak dlouho, aby tyto podmnožiny tvořily pouze příklady jedné třídy. Tento postup se také nazývá „top down induction of decision trees“ – TDIDT. Cílem je nalézt strom, který bude konzistentní s trénovacími daty. Přitom se dává přednost menším stromům, které jsou také jednodušší.

Algoritmus pracuje ve 3 krocích:

1. Zvol jeden atribut jako kořen stromu.
2. Rozděl data v tomto uzlu na podmnožiny podle hodnot tohoto atributu a přidej uzel pro každou podmnožinu.
3. Existuje-li uzel, pro který nepatří všechna data do téže třídy, pro tento uzel opakuj postup od bodu 1, jinak skonči.

Tento algoritmus bude pracovat pro kategoriální data.

Použití pro selekci atributů je pak velice jednoduché. Postupujeme od kořene až k listu. Přičemž atributy, které jsme potřebovali k rozhodování, pak tvoří vybranou množinu atributů. Jednotlivé úrovně stromu nemusí mít dotaz na stejný atribut a rovněž nemusí být cesta od kořene k listům stejně dlouhá – některé atributy lze vynechat. Ukazuje to i Obrázek 9. [18]

5 Experimenty

V této části práce ověřím schopnosti implementovaných algoritmů. Jak už jsem uvedl výše, metody selekce příznaků tvoří jen část řetězce klasifikace dat. Před samotnými experimenty je nutné provést několik kroků.

Prvním krokem je získání vhodných datových souborů. Data lze buď vygenerovat, nebo získat z internetu. Na internetu existuje mnoho specializovaných serverů zabývající se problematikou selekce příznaků a klasifikace dat. Některé weby umožňují také pří získávání datových souborů použití filtrů. Takže si můžeme jednoduše vybrat data určité velikosti z námi požadované oblasti, nebo s požadovaným druhem dat, jako např. spojitá nebo kategoriální.

Dalším krokem je úprava a předzpracování získaných dat. Existuje mnoho datových formátů a proto je nutné si data přizpůsobit pro svou aplikaci. Rovněž jsou v některých datech atributy, které jsou sice informačně důležité, avšak pro následující kroky nevhodné. Může se jednat např. o pořadové atributy jako číslo studie, číslo měření apod.

Nyní již konečně následuje selekce atributů. V tomto kroku dochází ke snížení počtu atributů. U každé metody budu provádět experimenty na vygenerovaných datech a posléze na datech reálných. Na vygenerovaných datech si ověřím, zda jsou metody správně naimplementovány a také jak silná je daná metoda. I když bude metoda správně fungovat, tak se může stát, že závislost dat neodhalí. Důvodem může být např. to, s jakými parametry bude metoda spuštěna. Díky těmto parametrům taky můžeme určit, jak moc chceme snížit počet atributů.

Jako poslední krok je porovnání, jaký vliv mělo odstranění atributů na výsledek klasifikace. Proto nejprve provedu klasifikaci na datech se všemi atributy a zaznamenám výsledky klasifikace. Pak pomocí mé aplikace provedu selekci příznaků. Nakonec opět provedu klasifikaci dat a porovnám, zda došlo ke zlepšení nebo ne.

5.1 Chi-square test

K vyhodnocení tohoto testu je potřeba znát kritické hodnoty pro jednotlivé stupně volnosti a pro vybranou hladinu významnosti. Tabulky jsem převzal z [11].

Výstupem testu je:

- seznam všech kontingenčních tabulek (v tomto případě je pouze jedna)
- seznam tabulek odhadovaných četností
- označení tabulek, u kterých je nutno sloučit řádky
- seznam všech kontingenčních tabulek, u kterých došlo ke sloučení (pokud takové existují)
- seznam všech tabulek odhadovaných četností, u kterých došlo ke sloučení (pokud takové existují)
- informace, že chi-square test nemohl být dokončen z důvodu nesplnění požadovaných vstupních podmínek
- zda existuje mezi atributy závislost
- aktuální sloupce, kterých se týká tato závislost
- vypočtené hodnoty chi-square testu
- jednotlivé kritické hodnoty
- seznam atributů (sloupců), které budou odstraněny

5.1.1 Experiment 1

Vstup: dataset1.xlsx

Počet atributů: 2 atributy

Popis: Pro první test jsem si vygeneroval datový soubor se 2 atributy – „Typ zařízení“ a „Uhlopříčka“. Jako typ zařízení jsem zvolil hodnoty *mobil*, *tablet* a *PC*. Uhlopříčka displeje daných zařízení je *3*, *7* a *15* palců. Datový soubor je schválně vytvořený tak, že atributy jsou maximálně závislé.

Volba hypotéz: H_0: Mezi atributy neexistuje závislost.

H_A: Atributy jsou závislé.

Parametry: Hladina významnosti 0,001 (velmi nízká)

Výstup: Výsledek testu ukazuje Obrázek 10.

Závěr: Podle předpokladů měl být 2. atribut silně závislý na 1. atributu. Vypočtená chi-square hodnota vyšla 120. Kritická hodnota byla 18,467. Vypočtená hodnota je vyšší než kritická hodnota, proto zamítám nulovou hypotézu ve prospěch hypotézy alternativní. Tedy platí, že mezi atributy existuje závislost. Proto dojde k odstranění druhého atributu, což je vidět i na výstupu testu.

```
Kontingencni tabulka 1
19 0 0
0 18 0
0 0 23

Tabulka odhadovanych cetnosti 1
6,02 5,7 7,28
5,7 5,4 6,9
7,28 6,9 8,82

Zavislost: Ano
Sloupce: 1-2
Chi-square hodnota: 120
Kriticka hodnota: 18,467

Sloupce k odstraneni: 2,
```

Obrázek 10: Výsledek chi-square testu

5.1.2 Experiment 2

Vstup: dataset2.xlsx

Počet atributů: 2 atributy

Popis: Další test nad „umělými“ daty jsem se snažil vytvořit tak, aby mezi atributy závislost nebyla. Opět jsem zvolil pro jednoduchost dva atributy a to „Věk osoby“ a „Počet dětí“. Protože tento test pracuje s diskrétními hodnotami, rozdělil jsem věk do několika intervalů a počet dětí také. Pro náhodné vygenerování počtu dětí jsem využil generátor náhodných čísel programu Excel. Takto by měla být zachována nezávislost hodnot mezi třídami.

Volba hypotéz: H_0: Mezi atributy neexistuje závislost.

H_A: Atributy jsou závislé.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Testová hodnota vyšla 13,666. Kritická hodnota byla 31,41.

Závěr: Zde není vypočtená hodnota vyšší než hodnota kritická a proto platí nulová hypotéza. Atributy tedy nejsou závislé. Proto také nedojde k odstranění žádných dat ve výstupním datovém souboru.

Předcházející dva experimenty měly za úkol demonstrovat závislosti mezi atributy. V případě prokázané závislosti měly být tyto atributy odstraněny. V dalších experimentech již budu provádět zároveň klasifikaci. Zhodnotím, jak velkou úspěšnost bude mít chi-square test na výslednou klasifikaci u zvolených datových souborů.

5.1.3 Experiment 3

Vstup: dataset3.xlsx

Počet atributů: 22 atributů + 1 klasifikační atribut

Popis: Pomocí generátoru náhodných čísel aplikace Excel jsem vygeneroval pro 20 atributů náhodná data z množiny <0;5>. Pro další 2 atributy jsem vytvořil závislost na prvním a druhém atributu pomocí přičtení hodnoty 1 resp. 2. Klasifikační sloupec tvoří podmínka, zda je součet předcházejících sloupců větší než 60. Takto jsou vytvořeny dvě klasifikační třídy. Aby aplikace Weka rozpoznala tato data jako kategoriální, přejmenoval jsem hodnoty na jejich jmenné vyjádření (0 = nula, …).

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Odstraněno 11 atributů.

Popis: Byl spuštěn test nad stejnými daty s nižší hladinou významnosti.

Parametry: Hladina významnosti 0,001 (velmi nízká)

Výstup: Odstraněny 2 závislé atributy.

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus/Data	Před FS	FS - 0,05	FS - 0,001
J48	67%	69%	67%
SimpleCart	63,4%	65,4%	63,4%

Tabulka 1: Úspěšnost klasifikace

Závěr: V první fázi experimentu došlo k odstranění 2 závislých atributů, což je v pořádku. Ačkoliv byly atributy generovány náhodně, bylo nalezeno i dalších 9 závislých atributů, které byly rovněž odstraněny. V druhé fázi experimentu se odstranily pouze 2 závislé atributy. Jak ukazuje tabulka, nepatrně lepších výsledků klasifikace bylo dosaženo při nastavené hladině významnosti na 0,05.

5.1.4 Experiment 4

Vstup: dataset4.xlsx [23]

Počet atributů: 22 atributů + 1 klasifikační atribut

Popis: Tato datová sada obsahuje popisy hub z rodiny Agaricus a Lepiota. Konkrétně jde o parametry klobouku, žáber, stonku, závoje a spórů. Je zde také uvedena lokalita výskytu těchto hub. Klasifikační třída zaznamenává, které houby jsou jedovaté a které jedlé. Data poskytl Jeff Schlimmer. Průzkum pochází z roku 1987.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Test nemohl být dokončen z důvodu nesplnění vstupních podmínek chi-square testu.

Vstup: dataset5.xlsx [24]

Počet atributů: 35 atributů + 1 klasifikační

Popis: Datová sada zaznamenává informace o sójových bobech v návaznosti na klasifikaci do 19 tříd podle nákazy bobů nebo jejich poškození. Jsou zaznamenány parametry rostlin, vliv počasí nebo také datum sběru. Soubor vytvořili Ming Tan a Jeff Schlimmer v roce 1988.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Test nemohl být dokončen z důvodu nesplnění vstupních podmínek chi-square testu.

Vstup: dataset6.xlsx [25]

Počet atributů: 10 atributů + 3 klasifikační atributy

Popis: Tento soubor obsahuje záznam slunečních erupcí. Zaznamenává jejich aktivitu v daném dni, aktivitu z předcházejícího dne, historické informace či informaci o velikosti erupční oblasti. Klasifikační třídy mapují erupce v dalších 24 hodinách. Jsou rozděleny na běžné, středně silné a silné erupce. Výzkum poskytl Gary Bradshaw v roce 1989.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Test nemohl být dokončen z důvodu nesplnění vstupních podmínek chi-square testu.

Klasifikace: Z důvodu nedokončení chi-square testů nemohla být vyhodnocena změna úspěšnosti klasifikace.

Závěr: V chi-square testu se vypočítávají tzv. kontingenční tabulky očekávaných četností. Jak jsem uvedl v teoretické části této práce, pro hodnoty v těchto tabulkách platí určité podmínky. Pokud tyto podmínky nejsou splněny, můžou být řádky těchto tabulek sloučeny. Minimální počet řádků po sloučení je omezen na 2. Jelikož v žádném z těchto tří případů nebyla tato podmínka splněna, nemohly být testy dokončeny.

5.1.6 Experiment 5

Vstup: dataset7.xlsx [26]

Počet atributů: 5 atributů + 1 klasifikační atribut

Popis: Zvolil jsem průzkum od Matěje Říhy z roku 2014. Zabývá se otázkou kouření. V průzkumu se zaznamenává četnost kouření, denně vykouřené cigarety, délku kouření, pohlaví a věk. Klasifikuje se pak, zda jedinec kouří nebo ne.

Data jsem musel upravit. Odpovědi u atributu „Značka cigaret“ byla formou textu a ne výběrem. Důsledkem by byla spousta tříd a nepřesností – např. 2 značky napsané vždy trochu jinak.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Došlo k odstranění 4 atributů, které byly shledány závislými na ostatních. Bylo tedy zjištěno, že existuje závislost mezi atributy "Jak často kouříte", "Kolik denně vykouříte cigaret" a "Jak dlouho kouříte". Čím osoba déle kouří, tím častěji kouří. Rovněž čím častěji kouří, tím více vykouří cigaret. Taktéž se frekvence kouření zvyšuje s věkem. Dle vypočtené chi-square hodnoty ale neexistuje závislost mezi intenzitou kouření a pohlavím.

Popis: Byl spuštěn test nad stejnými daty s nižší hladinou významnosti.

Parametry: Hladina významnosti 0,001 (velmi nízká)

Výstup: U tohoto testu došlo k ponechání atributu „Věk“, který se stal na zvolené hladině významnosti nezávislý s ostatními atributy.

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus/Data	Před FS	FS - 0,05	FS - 0,001
J48	97,65%	97,96%	97,96%
SimpleCart	98,74%	97,96%	97,96%
BayesNet	98,74%	97,95%	97,95%

Tabulka 2: Úspěšnost klasifikace

Závěr: Úspěšnost klasifikace byla velmi vysoká. U klasifikačního algoritmu J48 velice nepatrně pomohlo odstranění závislých atributů. Naopak výsledek u SimpleCart algoritmu se zhoršil, opět velice nepatrně. Provedl jsem tedy ještě klasifikaci s algoritmem BayesNet. Ten vykazoval rovněž zhoršení. Hladina významnosti na výsledek klasifikace vliv neměla.

5.2 Pearsonův korelační koeficient

5.2.1 Experiment 6

Vstup: dataset8.xlsx

Počet atributů: 11 atributů

Popis: Pro ověření tohoto testu jsem si vygeneroval pomocí generátoru náhodných čísel programu Excel náhodné hodnoty z intervalu 0 – 1. Vzhledem k hodnotám by atributy měly mít mezi sebou velmi nízkou závislost. Proto nastavím kritickou hodnotu rovněž velmi malou.

Parametry: Kritická hodnota nastavena na nízkou hodnotu 0,3.

Výstup: Nedošlo k odstranění žádného atributu.

Parametry: Kritická hodnota nastavena na velmi nízkou hodnotu 0,2.

Výstup: Nedošlo k odstranění žádného atributu.

Závěr: Kritické hodnoty byly nastaveny na velmi nízkou hodnotu. Při normálních experimentech se takto nízké běžně nenastavují. Došlo by k odstranění atributů s velmi nízkou závislosti a tedy ke zhoršení výsledné klasifikace. Ukazuje to i graf jednoho z dalších experimentů, viz Obrázek 11. Vzhledem k charakteru dat jsem si to ale mohl dovolit. Výstupy z testu byla nezávislost atributů potvrzena.

5.2.2 Experiment 7

Vstup: dataset9.xlsx

Počet atributů: 8 atributů

Popis: Pro tento experiment jsem si vygeneroval 7 závislých atributů. Závislosti se dají napsat pomocí následujících rovnic.

- $a_2 = -a_1$
- $a_3 = a_1 + 1$
- $a_4 = a_1^2$
- $a_5 = \sqrt{a_1}$
- $a_6 = a_1^2 + a_1 - 5$
- $a_7 = a_1^3$
- $a_8 = 1{,}1^{a_1}$

Parametry: Kritická hodnota nastavena na 0,7.

Výstup: Došlo k odstranění všech závislých atributů.

Parametry: Kritická hodnota nastavena na 0,8.

Výstup: Atributy a_2 – a_7 byly odstraněny. Atribut a_8 již odstraněn nebyl. Vypočtená hodnota testu 0,71 není vyšší než 0,8 a proto byl atribut ponechán.

Závěr: Na tomto testu byla ověřena jeho síla. U lineárních, kvadratických nebo kubických závislosti vyšla testová hodnota vyšší než 0,9. To značí velmi silnou závislost. U exponenciální funkce klesla vypočtená hodnota na 0,71. Stále se však jedná o silnou závislost.

Tyto dva experimenty měly za úkol demonstrovat závislosti mezi atributy. V případě prokázané závislosti tyto závislé atributy odstranit. Na dalších příkladech již budu provádět zároveň i klasifikaci. Zhodnotím, jak se změní úspěšnost klasifikace v závislosti na odstraněných atributech.

5.2.3 Experiment 8

Vstup: dataset10.xlsx [27]

Počet atributů: 18 atributů + 1 klasifikační atribut

Popis: Tento datový soubor obsahuje záznamy ze tří studií o modelování klimatu. Jednotlivé atributy tvoří 18 parametrů klimatických modelů. Klasifikační atribut ukazuje, zda simulace skončila zdárně nebo došlo k selhání. Studie pochází z Lawrence Livermore National Laboratory od sedmi autorů.

Datový soubor jsem upravil. Odstranil jsem atributy o pořadí studie a simulace.

Parametry: Kritická hodnota nastavena na 0,7.

Výstup: Došlo k odstranění 1 atributu.

Parametry: Kritická hodnota nastavena na 0,9.

Výstup: Nedošlo k odstranění žádného atributu.

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus/Parametr	Před FS	0,7
SMOreg	54,92%	54,92%
DecisionTable	66,74%	66,74%
M5Rules	66,28%	64,89%

Tabulka 3: Chyba klasifikace

Závěr: Vybrané algoritmy provedly klasifikaci s relativně velkou chybou. Odstranění jednoho atributu v případě SMOreg a DecisionTable algoritmů nemělo vliv na výslednou klasifikaci. U algoritmu M5Rules nepatrně klesla chyba klasifikace.

5.2.4 Experiment 9

Vstup: dataset11.csv [28]

Počet atributů: 10000 atributů + 1 klasifikační atribut

Popis: Jedná se o jeden z pěti testovacích souborů („Arcene“) v rámci Feature Selection Challenge. Cílem tohoto projektu je najít algoritmy pro selekci atributů, které poskytují mnohem lepší výsledky, než aktuálně známé. Testovací soubory se vyznačují velkým množstvím atributů.

Parametry: Kritická hodnota nastavena postupně na 0,2 – 0,99.

Výstup: Počet odstraněných atributů ukazuje Tabulka 4.

Parametr	Před FS	0,2	0,3	0,4	0,5	0,6
Odstraněných atributů	0	9831	9421	8389	7582	7171

Parametr	0,7	0,8	0,9	0,95	0,98	0,99
Odstraněných atributů	6921	6705	6318	5895	5151	4429

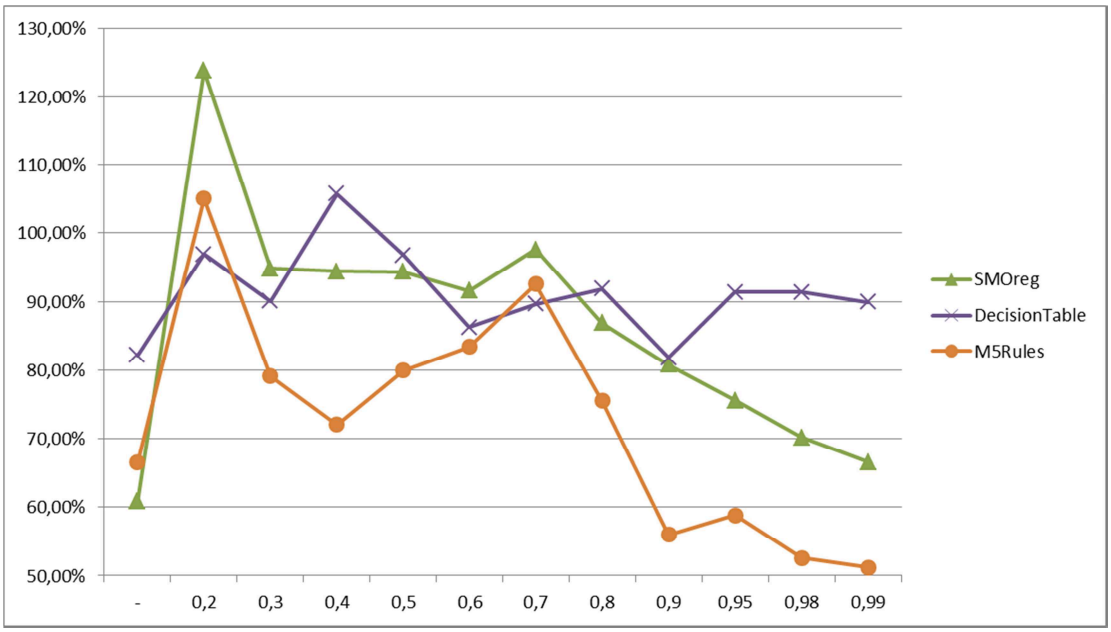
Tabulka 4: Počet odstraněných atributů

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus/Parametr	Před FS	0,2	0,3	0,4	0,5	0,6
SMOreg	60,80%	123,81%	94,82%	94,41%	94,34%	91,59%
DecisionTable	82,16%	96,86%	90,07%	105,68%	96,72%	86,25%
M5Rules	66,49%	104,96%	79,09%	72,06%	79,90%	83,42%

Algoritmus/Parametr	0,7	0,8	0,9	0,95	0,98	0,99
SMOreg	97,59%	86,86%	80,73%	75,57%	70,16%	66,55%
DecisionTable	89,65%	91,81%	81,82%	91,35%	91,36%	89,92%
M5Rules	92,50%	75,53%	56,01%	58,85%	52,59%	51,21%

Tabulka 5: Vliv parametru na chybu klasifikace



Obrázek 11: Grafické vyjádření vlivu parametru na chybu klasifikace

Závěr: Klasifikační algoritmy před odstraňováním atributů vykazovaly relativně velkou chybu. Je to proto, že testovací soubor obsahuje velké množství závislých atributů. Spustil jsem test pro parametry 0,2 – 0,9. Čím nižší parametr, tím se odstranilo více atributů. Z grafu je vidět, že hodnotu parametru nemá smysl volit nižší než 0,8. Zde dochází jednak ke zhoršení klasifikace a jednak k velkým výkyvům. Když se parametr blížil k hodnotě 0,9, úroveň chyby klasifikace se snižovala. Proto jsem otestoval ještě oblast 0,9 – 0,99. Se zvyšujícím se parametrem se snižovala chyba klasifikace. Nejlepších výsledků dosahoval algoritmus M5Rules. Z grafu je vidět tendence snížení chyby klasifikace pro hodnotu parametru blížící se k 1. Patrně by pro ještě vyšší hodnoty parametru než 0,99 dosáhly algoritmy dalších zlepšení.

5.3 Kruskal-Wallis test

5.3.1 Experiment 10

Vstup: dataset12.xlsx

Počet atributů: 9 atributů

Popis: Pro ověření Kruskal-Wallis testu jsem si vygeneroval 8 závislých atributů. Tyto závislosti se dají popsat následujícími rovnicemi.

- $a_2 = -2*a_1$
- $a_3 = a_1 + 100$
- $a_4 = a_1^2$
- $a_5 = a_1^3$
- $a_6 = a_1^4$
- $a_7 = a_1^2 + a_1^3$
- $a_8 = \log a_1$
- $a_9 = 1{,}3^{a_1}$

Parametry: Hladina významnosti 0,1

Výstup: Došlo k odstranění všech závislých atributů.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Došlo k odstranění všech závislých atributů.

Parametry: Hladina významnosti 0,001 (velmi nízká)

Výstup: Došlo k odstranění všech závislých atributů.

Závěr: Experiment ukázal sílu Kruskal-Wallis testu. Na zvolených hladinách významnosti odhalil všechny závislosti mezi atributy. Test vykazoval vysoké výstupní hodnoty. Neměl problémy ani s vyššími mocninnými funkcemi, s exponenciální funkcí či s logaritmickou funkcí.

5.3.2 Experiment 11

Vstup: dataset13.xlsx

Počet atributů: 20 atributů

Popis: Pro zjištění citlivosti testu jsem si vygeneroval pomocí generátoru náhodných čísel programu Excel náhodné hodnoty z intervalu 0 – 10000. Vzhledem k charakteru hodnot by atributy měly mít mezi sebou velmi nízkou závislost.

Parametry: Hladina významnosti 0,001 (velmi nízká)

Výstup: Nedošlo k odstranění žádného atributu.

Parametry: Hladina významnosti 0,05 (běžně používána)

Výstup: Nedošlo k odstranění žádného atributu.

Parametry: Hladina významnosti 0,1

Výstup: Nedošlo k odstranění žádného atributu.

Závěr: Dle předpokladů nebyly při testování odstraněny žádné atributy. Testové hodnoty vyšly vždy velmi nízké a proto i na hladině významnosti 0,1 zůstaly všechny atributy zachovány.

U dvou předcházejících experimentů jsem demonstroval citlivost testu u závislých a nezávislých atributů. V dalších experimentech budu provádět také klasifikaci výsledných souborů. Použiju stejné datové soubory jako u experimentů s Pearsonovým relačním koeficientem a porovnám, který z těchto dvou testů bude mít větší přínos pro výslednou klasifikaci.

5.3.3 Experiment 12

Vstup: dataset14.xlsx

Počet atributů: 18 atributů + 1 klasifikační atribut

Popis: Tento datový soubor obsahuje záznamy ze tří studií o modelování klimatu. Jednotlivé atributy tvoří 18 parametrů klimatických modelů. Klasifikační atribut ukazuje, zda simulace skončila zdárně nebo došlo k selhání. Studie pochází z Lawrence Livermore National Laboratory od sedmi autorů.

Datový soubor jsem upravil. Odstranil jsem atributy o pořadí studie a simulace.

Parametry: Hladina významnosti nastavena postupně na 0,1 a 0,001.

Výstup: Došlo k odstranění 2 atributů.

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus	Před FS	Po FS
SMOreg	54,92%	54,90%
DecisionTable	66,74%	66,74%
M5Rules	66,28%	63,17%

Tabulka 6: Chyba klasifikace

Závěr: Algoritmy SMOreg a DecisionTable nezaznamenaly prakticky žádné zlepšení klasifikace. Lépe si vedl M5Rules algoritmus, u kterého se zvýšila úspěšnost klasifikace oproti původnímu souboru o 3%.

Co se týče porovnání výsledků s Experimentem 8, algoritmus M5Rules dosahoval v minulém případě úspěšnosti 64,89% a nyní 63,17%. Je zde tedy vidět zlepšení. Další dva klasifikační algoritmy vykazovaly v Experimentu 8 a nyní stejné výsledky. Závěrem lze říci, že odstranění dalšího atributu bylo přínosné.

5.3.4 Experiment 13

Vstup: dataset15.csv [28]

Počet atributů: 10000 atributů + 1 klasifikační atribut

Popis: Jedná se o jeden z pěti testovacích souborů („Arcene“) v rámci Feature Selection Challenge. Cílem tohoto projektu je najít algoritmy pro selekci atributů, které poskytují mnohem lepší výsledky, než aktuálně známé. Testovací soubory se vyznačují velkým množstvím atributů.

Parametry: Hladina významnosti nastavena postupně na 0,1, 0,05 a 0,001

Výstup: Počet odstraněných atributů ukazuje Tabulka 7.

Parametr	Před FS	0,1	0,05	0,001
Odstraněných atributů	0	1690	1670	1628

Tabulka 7: Počet odstraněných atributů

Klasifikace: Byly spuštěny jednotlivé klasifikace, typ „Cross-validation“ s parametrem 10 folds. Algoritmům zůstaly přednastavené hodnoty.

Algoritmus/Parametr	Před FS	0,1	0,05	0,001
SMOreg	60,80%	61,89%	61,87%	61,59%
DecisionTable	82,16%	87,22%	87,22%	87,22%
M5Rules	66,49%	60,21%	60,21%	65,53%

Tabulka 8: Chyba klasifikace

Závěr: Před provedením klasifikace dosahovaly klasifikační algoritmy celkem velkých chyb. Spustil jsem test s nastavenými hladinami významnosti 0,1, 0,05 a 0,001. Jak ukazuje Tabulka 7, rozdíl v počtech odstraněných atributů byl velice malý. Původní soubor obsahoval 10000 atributů a zde byl rozdíl odstraněných atributů mezi nejnižší a nejvyšší hladinou významnosti pouze 62. Tento výsledek již předem naznačuje, že nejspíše nebude velký rozdíl v úspěšnosti klasifikace v závislosti na jednotlivých hladinách významnosti. Tyto předpoklady potvrzují jak SMOreg, tak DecisionTable algoritmus. U M5Rules algoritmu však došlo ke znatelnému zhoršení – přes 5% (u 0,001). Pokud však zvolíme vyšší hladinu významnosti, tak tento algoritmus dosahuje nejlepších výsledků ze všech tří klasifikačních algoritmů. Při porovnání před a po FS jako jediný dosahuje tento algoritmus lepšího výsledku, a to o více než 6%.

Porovnání tohoto experimentu s experimentem 9 není jednoduché. Předcházející test odstraňoval velké množství atributů, tento mnohem menší. SMOreg a DecisionTable algoritmy vykazovaly lepší výsledky zde, M5Rules zase v experimentu 9. V tomto experimentu dosáhl i nejlepšího výsledku klasifikace na zvolený datový soubor, a to chyby jen 51,21%.

6 Závěr

V páci jsem se zabýval metodami pro výběr atributů. Uvedl jsem, proč je vhodné snižovat dimenzionalitu datového souboru, a rozdělil jsem tento problém na dva odlišené postupy řešení. Nastínil jsem rovněž pojem klasifikace dat.

Přehled algoritmů jsem rozdělil do tří částí na metody typu „filters“, „wrappers“ a „embedded“. Naimplementoval jsem metody chi-square test, Kruskal–Wallis test a také korelační metodu, která pracuje s Pearsonovým korelačním koeficientem. U každé této metody jsem provedl několik experimentů na mnou vygenerovaných a také na reálných datech.

V experimentech u chi-square testu jsem po provedení FS zaznamenal v některých případech mírné zlepšení úspěšnosti klasifikace, v jiných zase mírně zhoršení. Z tohoto pohledu se dá algoritmus doporučit. Algoritmus je však citlivý na rovnoměrné rozložení dat v datovém souboru. Pokud tomu tak není, algoritmus to při svém běhu detekuje a test se nedokončí z důvodu nesplnění vstupních podmínek testu. Stalo se to takto u třech datových souborů. Algoritmus při svém běhu buduje tzv. kontingenční tabulky. Při velkých datových souborech je proto náročný na operační paměť. Mým cílem do budoucna je kód optimalizovat, aby se mohl vyrovnat i s velkými datovými soubory, kde jsem zaznamenal paměťové problémy.

U experimentů s Pearsonovým relačním koeficientem jsem provedl test, kdy jsem nastavil hodnotu parametru napříč celým možným rozsahem. Potvrdilo se, že je vhodné volit tento parametr vysoký nebo velice vysoký. Při těchto hodnotách zaznamenal již algoritmus zlepšení vůči stavu před FS. Pro nižší hodnoty parametru, kdy byla korelace slabá, se choval nepřesvědčivě. Zde odstraňoval velké množství atributů, což způsobilo zhoršení výsledků.

Použití Kruskal-Wallis testu je časově náročnější, než výpočet předešlé korelace. Test však při experimentech vykazoval rovnoměrnější výsledky. Pokud by tedy nehrála časová náročnost až tak velkou roli, zvolil bych si tento test.

7 Použitá literatura

1. *Örebro university.* [Online] [Citace: 16. 7. 2014] http://oru.diva-portal.org/smash/get/diva2:567115/FULLTEXT01.pdf.

2. *Center for Machine Perception, ČVUT.* [Online] [Citace: 17. 7. 2014] cmp.felk.cvut.cz/~hlavac/Public/TeachingLectures/UceniBezUcitele.pdf

3. Weka (machine learning). *Wikipedia, the free encyclopedia.* [Online] [Citace: 17. 7. 2014] http://en.wikipedia.org/wiki/Weka_(machine_learning)

4. The ID3 Algorithm. *University Of Florida.* [Online] [Citace: 16. 7. 2014] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

5. J48 decision tree. *Data Mining with R.* [Online] [Citace: 16. 7. 2014] http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree

6. Bayesova věta. *ScienceGraph.* [Online] [Citace: 16. 7. 2014] http://cs.sciencegraph.net/wiki/Bayesova_věta

7. SMOreg. *Pentaho.* [Online] [Citace: 16. 7. 2014] http://wiki.pentaho.com/display/DATAMINING/SMOreg

8. M5Rules. *Department of Computer Science and Engineering, Indian Institute of Technology Bombay.* [Online] [Citace: 16. 7. 2014] http://www.cse.iitb.ac.in/infolab/cep/day7/lab/tools/weka-3-5-3/doc/weka/classifiers/rules/M5Rules.html

9. **Briš, Radim a Litschmannová, Martina.** *STATISTIKA I.* 2004.

10. How to use a chi squared calculator to analyze multiple variables. *SimaFore.* [Online] [Citace: 15. 7. 2014] http://www.simafore.com/blog/bid/108131/How-to-use-a-chi-squared-calculator-to-%20%20%20analyze-multiple-variables

11. 1.3.6.7.4. Critical Values of the Chi-Square Distribution. *Engineering Statistics Handbook.* [Online] [Citace: 7. 7. 2014] http://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm

12. 5 simple steps to apply chi-square test for business analytics. *SimaFore.* [Online] [Citace: 15. 7. 2014] http://www.simafore.com/blog/bid/54885/5-simple-steps-to-apply-chi-square-test-for-business-analytics

13. Korelační koeficient. *BATCOS.* [Online] [Citace: 16. 7. 2014] http://athena.zcu.cz/kurzy/spne/000/HTML/40/

14. PEARSON (funkce). *Office.* [Online] [Citace: 16. 7. 2014] http://office.microsoft.com/cs-cz/excel-help/pearson-funkce-HA102752907.aspx

15. *Institut biostatiky a analýz.* [Online] [Citace: 16. 7. 2014] http://www.iba.muni.cz/esf/res/file/bimat-prednasky/biostatistika-pro-matematickou-biologii/BpMB-11.pdf.

16. Pearsonův korelační koeficient. *ABZ slovník cizích slov.* [Online] [Citace: 16. 7. 2014] http://slovnik-cizich-slov.abz.cz/web.php/slovo/pearsonuv-korelacni-koeficient

17. *Ústav automatizace a informatiky.* [Online] [Citace: 16. 7. 2014] www.uai.fme.vutbr.cz/~jdvorak/vyuka/es/MachLearn.ppt

18. *Vysoká škola ekonomická v Praze.* [Online] [Citace: 16. 7. 2014] http://sorry.vse.cz/~berka/docs/izi456/kap_5.1.pdf

19. *Center for Computational Biology - Ecole des Mines de Paris.* [Online] [Citace: 17. 7. 2014] http://cbio.ensmp.fr/~jvert/svn/bibli/local/Liu2004comparative.pdf

20. Wrappers for feature subset selection. *ResearchGate.* [Online] [Citace: 16. 7. 2014] http://www.researchgate.net/publication/221996228_Wrappers_for_feature_subset_selection

21. *Masarykova univerzita, Institut biostatiky a analýz.* [Online] [Citace: 16. 7. 2014] http://www.iba.muni.cz/esf/res/file/bimat-prednasky/analyza-a-klasifikace-dat/AKD-08.pdf

22. *Computer Science and Engineering, Michigan State University.* [Online] [Citace: 16. 7. 2014] www.cse.msu.edu/~cse802/Feature_selection.pdf

23. *UCI Machine Learning Repository.* [Online] [Citace: 9. 7. 2014] https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/

24. *UCI Machine Learning Repository.* [Online] [Citace: 9. 7. 2014] https://archive.ics.uci.edu/ml/machine-learning-databases/soybean/

25. *UCI Machine Learning Repository.* [Online] [Citace: 9. 7. 2014] https://archive.ics.uci.edu/ml/machine-learning-databases/solar-flare/

26. Kouření (výsledky průzkumu) | Vyplňto.cz - řešení i pro Váš internetový průzkum. *Vyplňto.cz.* [Online] 9. 7. 2014 http://www.vyplnto.cz/realizovane-pruzkumy/40044/

27. *UCI Machine Learning Repository.* [Online] [Citace: 12. 7. 2014] https://archive.ics.uci.edu/ml/machine-learning-databases/00252/

28. Feature Selection Challenge - Datasets. *Feature Selection Challenge.* [Online] 12. 7. 2014 http://www.nipsfsc.ecs.soton.ac.uk/datasets/